Informe de Análisis y Respuesta a Múltiples Incidentes de Seguridad

1. Resumen Ejecutivo

El presente informe detalla el análisis de cinco eventos de seguridad detectados en la empresa SecureWeb Ltd. Los eventos han sido clasificados según la respuesta requerida (mitigación, prevención o ambas), se han propuesto medidas técnicas específicas para cada uno y se ha establecido una prioridad de intervención basada en el impacto potencial y la facilidad de explotación. Se concluye que tres de los eventos detectados requieren una acción inmediata de mitigación, mientras que los otros dos, aunque graves, pueden ser abordados con medidas de prevención programadas.

2. Clasificación y Propuesta Técnica por Evento

A continuación, se clasifican y proponen medidas de mitigación y prevención para cada evento.

- **Evento: Múltiples intentos fallidos de login desde la misma IP.**
    - Clasificación: Ambos: Mitigación y Prevención.
    - Mitigación: Implementar un retraso progresivo en los intentos de login, aumentando el tiempo de espera después de cada fallo.
    - Prevención: Configurar un límite de intentos (ej. 5) y bloquear temporalmente la cuenta o la dirección IP de origen.
- **Evento: Inyección de código JavaScript en los campos de comentarios del blog.**
    - Clasificación: Ambos: Mitigación y Prevención.
    - Mitigación: Desinfectar la salida de los campos de comentarios en el *front-end* utilizando una librería como DOMPurify para neutralizar el código malicioso.
    - Prevención: Implementar validación de entrada en el servidor para rechazar datos que contengan etiquetas HTML o *scripts*.
- **Evento: Acceso no autorizado al *endpoint* /admin/export sin validación de rol.**
    - Clasificación: Ambos: Mitigación y Prevención.
    - Mitigación: Deshabilitar temporalmente el acceso a la ruta /admin/export o restringirlo a direcciones IP de confianza.

    - Prevención: Implementar un Control de Acceso Basado en Roles (RBAC) en el *backend* para asegurar que solo los usuarios con el rol de administrador puedan acceder al recurso.

- **Evento: Uso de una librería JavaScript con una vulnerabilidad conocida no parcheada.**
    - Clasificación: Prevención.
    - Mitigación: N/A. Este es un riesgo latente, no un incidente activo, por lo que no requiere una mitigación inmediata.
    - Prevención: Utilizar herramientas de Análisis de Composición de Software (SCA) para escanear y monitorear las librerías de terceros. Programar la actualización de la librería vulnerable.
- **Evento: Fallo en el *backend* que expone mensajes de error con estructura de base de datos.**
    - Clasificación: Ambos: Mitigación y Prevención.
    - Mitigación: Configurar el servidor de producción para desactivar los mensajes de error detallados y registrar la información en los *logs* internos.
    - Prevención: Implementar un manejo de errores personalizado en el *backend* que muestre mensajes genéricos al usuario, ocultando la estructura interna de la base de datos.

3. Priorización de la Intervención y Justificación

La siguiente lista presenta la prioridad de acción de los eventos, ordenados de mayor a menor urgencia, basándose en su nivel de exposición, facilidad de explotación e impacto potencial.

1. Acceso no autorizado al *endpoint* /admin/export (Prioridad Crítica): Este es un incidente activo que ya ha permitido el acceso no autorizado a datos sensibles. La facilidad de explotación es alta, ya que solo requiere la navegación a un *endpoint* público, y el impacto es máximo, con riesgo de filtración de información crítica.
2. Inyección de código JavaScript (XSS) (Prioridad Alta): Este evento es un incidente activo de explotación. Un atacante podría robar sesiones de usuarios o redirigirlos a sitios maliciosos, comprometiendo la seguridad de múltiples usuarios simultáneamente.

3. Fallo que expone mensajes de error del *backend* (Prioridad Alta): Aunque no es un incidente de explotación activa, la revelación de la estructura de la base de datos facilita enormemente futuros ataques, como la Inyección SQL. La mitigación es urgente para eliminar esta información valiosa para un atacante.
4. Múltiples intentos fallidos de login (Prioridad Media): La amenaza es un ataque de fuerza bruta potencial, lo que representa un riesgo significativo de compromiso de cuentas. La mitigación es necesaria para prevenir un incidente mayor.
5. Uso de librería JavaScript con vulnerabilidad conocida (Prioridad Baja): Este es un riesgo latente y no un incidente activo. La vulnerabilidad no ha sido explotada hasta el momento, lo que permite un abordaje programado y no de emergencia. La prioridad es corregirla de forma proactiva para evitar una futura explotación.